Proceedings of Acivs 2011, Ghent, Aug. 22-25 2011

Robust Visual Odometry using Uncertainty Models

David Van Hamme[1,2], Peter Veelaert[1,2], and Wilfried Philips[2]

[1] University College Ghent (Vision Systems)
[2] Ghent University/IBBT (IPI)

Abstract. In dense, urban environments, GPS by itself cannot be relied on to provide accurate positioning information. Signal reception issues (e.g. occlusion, multi-path effects) often prevent the GPS receiver from getting a positional lock, causing holes in the absolute positioning data. In order to keep assisting the driver, other sensors are required to track the vehicle motion during these periods of GPS disturbance. In this paper, we propose a novel method to use a single on-board consumer-grade camera to estimate the relative vehicle motion. The method is based on the tracking of ground plane features, taking into account the uncertainty on their backprojection as well as the uncertainty on the vehicle motion. A Hough-like parameter space vote is employed to extract motion parameters from the uncertainty models. The method is easy to calibrate and designed to be robust to outliers and bad feature quality. Preliminary testing shows good accuracy and reliability, with a positional estimate within 2 metres for a 400 metre elapsed distance. The effects of inaccurate calibration are examined using artificial datasets, suggesting a self-calibrating system may be possible in future work.

1 Introduction

The current generation of GPS navigation systems is insufficiently reliable in urban conditions. Obstacles along and over the road (e.g. tall buildings, overpasses, bridges) often stand in the way of the four-way satellite link that is required for a positional fix. In these situations, the GPS unit temporarily loses track of position. However, it is sufficient to be able to accurately track the relative motion of the vehicle over a short distance to keep a reasonable estimate of the absolute position.

In this paper we present a relative motion tracking approach based on a single forward-facing camera. There is a tendency in the automotive industry to equip consumer vehicles with cameras to perform a number of driver assistance tasks. Traffic sign recognition and lane departure warning are two examples that can readily be found on the options list of many new cars. The calculation of the ego-motion of a moving platform from images captured from the platform itself is called visual odometry.

A solid mathematical basis for visual odometry was laid in the past 20 years ([1, 10, 11, 13–15]). More recent works focus on how to bring these concepts into

practice. The approaches can be roughly divided into three categories based on the optics used. A first approach uses stereo cameras, with influential works by Obdržálek *et al.*, 2010 [16], Kitt *et al.*, 2010 [8], Comport *et al.*, 2007 [6], Konolige *et al.*, 2007 [9] and Cheng *et al.*, 2006 [5]. A second approach is the use of an omnidirectonal camera, as in Scaramuzza *et al.*, 2009 [17], Tardif *et al.*, 2008 [18]. The third category consists of methods that use a single, non-panoramic camera. Significant efforts include Campbell *et al.*, 2005 [4] and Mouragnon *et al.*, 2009 [12], Azuma *et al.*, 2010 [2].

Both stereo vision and panoramic cameras boast obvious advantages for solving the visual odometry problem. However, such setups are impractical for use on consumer vehicles, as the stereo methods require precise and repeated calibration and the omnidirectional camera cannot be mounted inconspicuously. We will therefore focus on a solution that involves a single, standard consumer camera. Among the methods cited above, good performance has been demonstrated on indoor test sequences, and in some cases even in controlled outdoor environments. However, the application on consumer vehicles poses new challenges. As road speeds increase, the number of available feature correspondences is reduced, and their displacements in the image can become very large. A third problem is the presence of outliers: other traffic will cause feature correspondences that are of no use to calculate visual odometry.

These unsolved problems lead us to propose a novel method for a single, forward-facing camera. Central to the method is the backprojection of image features to the world ground plane, and the uncertainties associated with this backprojection. A Hough-like voting scheme is implemented to track the consistent motion in the uncertainty models of the ground plane features. Careful modelling of the uncertainties both in the backprojection and in the motion parameters, coupled with a robust voting algorithm, allow us to reliably extract the ego-motion from a calibrated system. Experiments on real data show accuracy within 2 metres after an elapsed distance of 400 metres. The details of the method will be further explained in section 3.

2 Camera model

To accurately model the uncertainties of the backprojection in an elegant way, it is important to choose an appropriate camera model and carefully define the used coordinate systems. In our application, we are working in three different coordinate systems: 3D world coordinates, 3D camera coordinates, and 2D image coordinates. The 3D world coordinate system is chosen as a right-handed system with the origin on the road surface directly below the center of rotation of the vehicle. Note that this means that the world axes remain tied to the vehicle, and vehicle motion manifests itself as moving texture on a fixed plane. The world Z-axis is taken perpendicular to the ground plane, the Y-axis parallel to the straight-ahead driving direction of the vehicle. The camera coordinate system is also a right-handed system, and is similarly aligned. Its origin is in the center of projection of the camera, its Y-axis points in the viewing direction along

the principal axis, and the X- and Z-axis are aligned with the horizontal and vertical image sensor directions respectively. While this deviates somewhat from the conventional definitions, it facilitates calibration, as will be explained shortly. Finally, the 2D image coordinate system is defined with the origin in the top left corner of the image, the X-axis pointing right and the Y-axis pointing down. Choosing the axes this way offers some advantages from a programming point of view, as it better reflects image data organization in memory.

To describe the perspective projection from 3D world coordinates to 2D image coordinates, we will use an undistorted pinhole camera model. Let $\mathbf{x} = [x\ y\ w]^T$ denote the 2D image point in homogeneous coordinates, and $\mathbf{X} = [X\ Y\ Z\ 1]^T$ the corresponding point in homogeneous 3D world coordinates. The projection of $\mathbf{X}$ onto $\mathbf{x}$ is then given by:

$$\mathbf{x} = \mathbf{C}\left[\mathbf{R}|\mathbf{t}\right]\mathbf{X}. \tag{1}$$

In the above expression, $\mathbf{C}$ is the upper triangular intrinsic camera matrix as described in Hartley and Zisserman, 2004 [7], consisting of the horizontal and vertical scaling components α_x and α_y and the image coordinates of the principal point (x_0, y_0), multiplied by a substitution matrix arising from our non-standard definition of image axes:

$$\mathbf{C} = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{2}$$

$[\mathbf{R}|\mathbf{t}]$ is the rotation matrix $\mathbf{R}$ that aligns the world axes with the camera axes, augmented by the 3D translation vector $\mathbf{t}$ between their origins. The resulting 4x3 matrix is the projection matrix that maps homogeneous 3D world coordinates onto 2D image coordinates. Because we will only consider points in the world ground plane, $Z = 0$ for all points and the projection matrix reduces to a 3x3 homography matrix $\mathbf{H}$:

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \mathbf{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{3}$$

In our case, we are interested in the inverse transformation: we want to project image points onto the world ground plane. This projection is characterized by the inverse of $\mathbf{H}$:

$$\mathbf{X} = \begin{bmatrix} X \\ Y \\ W \end{bmatrix} = \mathbf{H}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{4}$$

If lens distortion has to be taken into account, a distortion function is applied to the left hand side of equations (1) and (3). We will consider the distortion to be rectified in advance, as can easily be done using standard methods [3].

Let us take a closer look at the components of equation (1). The camera matrix $\mathbf{C}$ can be obtained using standard calibration methods as described in Bouguet, 1999 [3], and will remain constant for a fixed-zoom camera.

The vector $\mathbf{t}$ determines the offset between the origin of the world axes and the origin of the camera axes. We can measure $\mathbf{t}$ as part of the extrinsic calibration process, as it changes only very slightly with vehicle motion and load.

The rotation matrix $\mathbf{R}$ can be constructed as a series of three rotations, each along one of the axes. The most common conventions for defining these rotations are through heading, pitch and roll (Z-, X-, and Y-axis rotations) or Euler angles (usually Z-X-Z). We will use the heading, pitch and roll configuration. Due to our non-standard choice of world and camera axes, these three angles can be measured relatively easily in world coordinates. However, only the heading can be assumed to remain constant while the vehicle is moving, as suspension movement does not affect this direction of the vehicle in any significant way. The other rotations however, pitch and roll, have a range of freedom around the calibration values obtained while stationary, as they are affected by the suspension. This range of motion will be the key challenge to overcome, as will be explained in section 3. The ground plane projection of a sample frame while stationary is shown in figure 1.

Fig. 1. Example of a test frame (left), and its ground plane backprojection (right).

3 The proposed method

The first step in our proposed method is the backprojection of Harris corners to the world ground plane, using equation (4). The advantage of using backprojected ground features is that the method will still produce useful motion information when there are only few feature correspondences, due to the reduction in degrees of freedom. However, in order to accurately backproject the Harris corners, we need the immediate pitch and roll angles for each frame. These angles are not precisely known, they are only defined within an interval around the calibration values obtained at rest (cfr. section 2). This range of possible suspension angles defines a region of possible backprojected locations in the world plane. Due to the trigonometric functions in the rotation matrix $\mathbf{R}$, this region will not strictly be convex, but as the variation in angles is small, it can be closely approximated by a tetragon. These uncertainty regions on the

world plane arising from the range of angles in the camera perspective will be referred to as **Perspective Uncertainty Tetragons** (PUTs). Note that there is a PUT for every Harris corner detected in the camera image.

The PUTs are easily calculated: a rotation matrix **R** is constructed for each of the four combinations of extremal pitch and roll values. This results in four different backprojections of each Harris corner, yielding the four corners of the tetragon for that feature.

It should be noted that any inaccuracies in the feature detector can also be modelled into the PUTs. Suppose for example that the chosen feature detector is known to have poor localization, producing features in positions that can be off by one pixel from the actual point of interest. One way to account for this could be to calculate the PUTs for all possible extremal pixel coördinates of the actual feature, distributed in an area around the feature detector output, and then taking the union of the PUTs to obtain the uncertainty on the world plane position. In our application, the Harris corners are assumed to be accurate within half a pixel in both the horizontal and vertical image dimension.

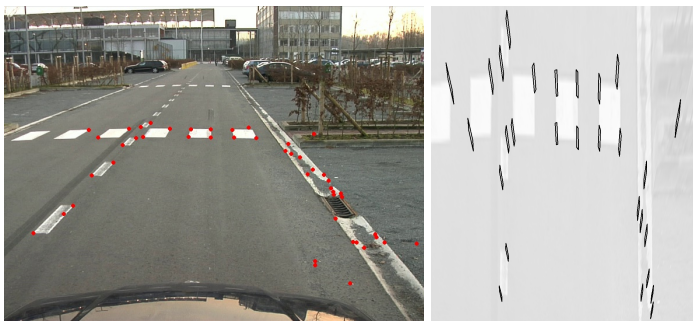

Fig. 2. Harris corners in the camera view (left), and close-up of some of their associated PUTs (right).

Figure 2 shows the PUTs for an example frame in a test sequence. The left image shows a camera frame in which Harris corners were detected. The background of the right image is the backprojection of the camera frame based on the average angles, with the PUTs of the Harris corners drawn on top of it. Realistic limits on the pitch and roll angles can be established by analysis of test videos in which the vehicle is accelerating, braking or cornering hard. We can see from the shape and size of the PUTs in figure 2 that pitch is the main factor contributing to the uncertainty (vertical elongation), while the effect of roll remains limited.

The problem we need to solve, is how to extract accurate translation and rotation parameters from correspondences between the features of consecutive frames when the position of the features is only known up to a region (the PUTs). Additionally, the method must be robust to outliers. Features that do

not correspond to object in the ground plane, or that belong to other traffic, will exhibit inconsistent motion and should have as little influence as possible.

To solve the above problem we propose a simple, transparent voting mechanism that can extract the incremental trajectory in real-time. In a first step, we must establish possible feature correspondences between consecutive frames. Let us assume that we know the exact ground plane position of the previous frame's features. As we stated earlier, the origin of the world coordinate system remains affixed to a point below the center of rotation of the vehicle. This means that one frame ahead, the features (which are static in the real world) will have new coordinates in the ground plane. The new coordinates depend on the speed and steering angle between the previous two frames, and on the acceleration and steering input of the driver between the previous and the current frame. The driver inputs are the unknowns that we want to determine. However, the acceleration is bound by the maximum torque of the engine and by the maximum retardation allowed by the braking system, and the steering input is limited by the ratio of the steering rack coupled with the maximum speed at which the driver is physically able to twirl the steering wheel. These bounds can be established from the specification of the vehicle and a simple experiment in which a person tries to turn the steering wheel from lock to lock as fast as possible.

The bounds on driver input are also included in our uncertainty model. For each known feature position from the previous frame, the range of possible driver inputs delimits a region in the world plane that represents this feature's possible world plane positions in the current frame. Again, this region is not strictly convex due to the rotation component, but is closely approximated by a tetragon. We will call this type of tetragon a **Motion Uncertainty Tetragon** (MUT), as it arises from the uncertainty on the vehicle motion. An example of a set of MUTs is shown in figure 3. Note that features in close proximity to the vehicle have a narrower MUT than distant features.

The MUT of a feature position is calculated by displacing the feature along four circle segments, representing the four extremal combinations of possible speed and steering angle. The circle segments are an approximation of the real trajectory, as they assume a constant speed and steering angle over the inter-frame interval. In reality, the bend radius will change continuously in this interval, but the errors introduced by this approximation are small.

The uncertainty of the feature detection and backprojection is now modelled in the PUTs and the uncertainty on the predicted displacement of the vehicle is modelled in the MUTs. The PUTs correspond to features in the current frame's camera view, while the MUTs correspond to features in the previous frame's world plane. The problem of finding correspondences between the previous frame and the current frame is thereby reduced to finding overlap between MUTs and PUTs.

Once the possible feature correspondences have been established, the second problem is how to extract the correct motion parameters (i.e. speed differential and steering angle differential) from these correspondences. Once we know these

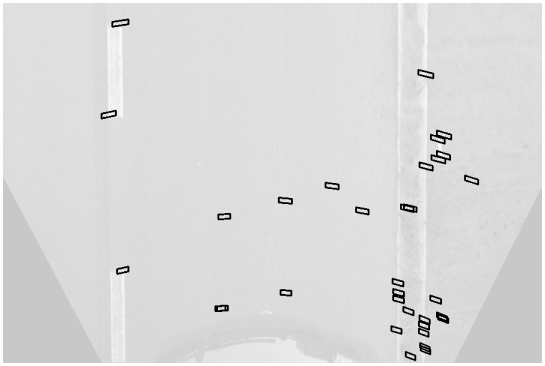

Fig. 3. Example of MUTs.

differentials, we can reconstruct the trajectory of the vehicle between the two frames.

The MUTs are essentially projections of a region of translation-rotation parameter space onto the ground plane. The overlap of the MUTs with the PUTs gives us information about which parameter combinations are plausible according to the observed features. Although the MUTs have slightly different sizes and shapes based on their location, their boundaries correspond to the same extremal values of speed and steering angle, and as such every MUT is a deformation of the same rectangular patch in parameter space. When a PUT overlaps with an MUT, this is evidence that the region of overlap in the MUT contains plausible vehicle motion parameters according to one of the features. We can state that the overlap expresses a vote for this region in parameter space. When we sum the region votes for all areas of PUT-MUT overlap, we obtain a measure of plausibility for every rotation-translation combination in the parameter space patch. Evidence will concentrate on those combinations that agree with the majority of the observations. This is similar in concept to Hough-based shape detection algorithms, where the shapes are found as peaks in a voted parameter space. This type of voting method has the advantage that it provides some robustness against outliers, as the contributions from bad features (e.g. features caused by other traffic or by objects not in the ground plane) will not typically have a common intersection, and as such tend to manifest themselves as noise spread out over the parameter space. Figure 4 shows the typical overlapping of MUTs and PUTs.

To perform the parameter space vote in practice, we normalize every MUT to a rectangle of predefined size and represent its overlap with one or more PUTs by a binary image of this predefined size. The horizontal axis of the normalized image is a slightly nonlinearly stretched representation of the rotation differential axis, while the vertical axis is the speed differential axis. Summing the images of all normalized MUTs is essentially the same as summing region votes in the discretized parameter space. An example of a sum image is shown in figure

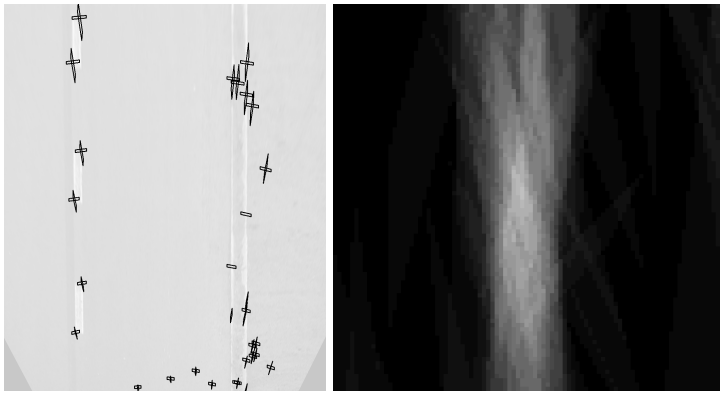

Fig. 4. Overlap between PUTs and MUTs, and corresponding consensus image.

4. We call this the **consensus image**, as it is a graphic representation of the parameter consensus between all features. In the example shown, we see that the votes concentrate on an area slightly below center. This translates to a slight deceleration while driving straight ahead.

We should note that in order for the evidence to concentrate in the consensus image, we need a horizontal distribution of features over the image as equal as possible. If Harris corners are only found on the right side of the driving direction, their PUTs will all be skewed in the same direction, causing their intersections to be large. Contributions of corners from the other side of the driving direction will yield much smaller intersections, and therefore a better concentration of evidence.

4 Results

The proposed method was tested on a trajectory on the parking lot of our campus. The trajectory is a figure of eight with a length of 431 metres. The reconstructed trajectory is shown in figure 5. The backprojections of the entire video frames are used as a background. The comparison with the ground truth of the trajectory is also shown in figure 5, with an aerial photo as background. The positional error at the end of this trajectory is 2.14 metres, while the rotational error is 7.67 degrees. When coupled to an offline map, this is sufficiently accurate to establish the road that the vehicle is travelling on. This proves the validity of the concept for filling in gaps in the GPS reception of a couple of hundred metres. However, more testing is required to evaluate robustness and the effect of other traffic (which was absent during the test run).

To evaluate the sensitivity of the method to calibration errors, the method was also tested on artificial data. A camera trajectory of 150 metres consisting of 2 sharp and 2 shallow bends above a checkerboard pattern was rendered and evaluated by the proposed method. The use of a rendered video provides

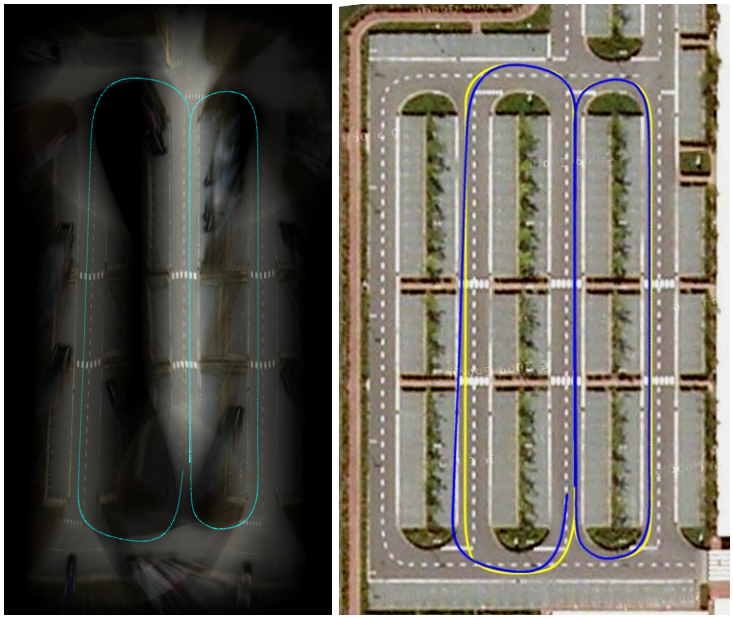

Fig. 5. Reconstructed trajectory and surroundings according to our system (left), compared to ground truth (yellow) on aerial photo.

the advantage that the calibration angles and trajectory are exactly known. An example of a frame from the artifical set is shown in figure 6. Figure 7 shows the results of our method compared to the ground truth, shown in black.

The red trajectory in figure 7 is the output of our method for correct calibration values. Positional and rotational errors are given in table 1. The green trajectory shows the result when the roll angle would be off by 2 degrees, for example due to a calibration error. Two distinct effects are visible. Firstly there is an overall rotational bias that manifests itself in both left and right bends. Secondly, the right bends get truncated while the left bends get elongated. Still, for our trajectory with equal bends left and right, the positional and rotational errors are relatively small. This shows that our method is reasonably robust against roll miscalibration. The blue trajectory in figure 7 results from a 2 degree error in the pitch calibration. The effect is significant: every displacement gets severely underestimated. Rotational accuracy is still good, but the positional error quickly becomes very large. We can conclude that the method is fairly sensitive to errors in pitch calibration. Finally, the teal trajectory shows the effect of a 2 degree error in heading. Predictably, there is a strong rotational bias, both in the bends and on the straight segments. The end position is off by a very wide margin, and likewise the rotation. Clearly, the method is also sensitive to heading calibration errors.

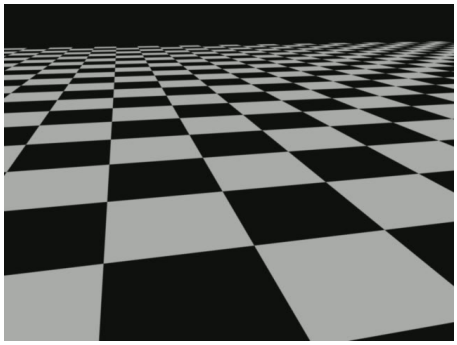

Fig. 6. Example frame of artificial test sequence.

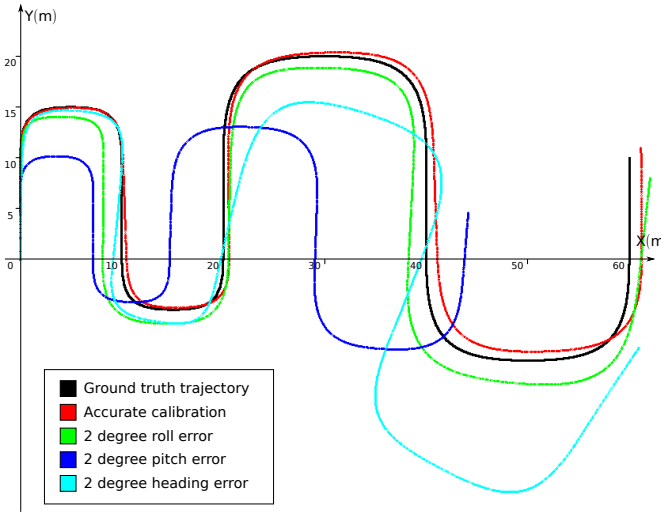

Fig. 7. Results for artificial test data, showing the effect of various calibration errors.

However, it should be noted that when an offline map is present, these systematic errors in position and rotation are easily detected, and the different effects of errors in the different angles should enable us to identify which calibration values are off. This in turn opens the road to a self-correction system, which would be a great asset for consumer applications.

	Positional error (m)	Rotational error (degrees)
Exact calibration	1.51	0.44
2 Degree roll error	2.83	-6.37
2 Degree pitch error	16.78	-4.64
2 Degree heading error	18.79	-33.40

Table 1. Positional and rotational errors for different miscalibrations.

5 Conclusion

We proposed a novel method for calculating visual odometry. The method is shown to work on real data, yielding a positional error of only around 2 metres on trajectory over 400 metres long. However, robustness has to be tested further on real-world data sets with other traffic present, and the algorithm should be connected to an offline map to keep drift under control for longer sequences. Despite the early stage of development though, the technique has already been proven accurate enough to pinpoint the lane in which the vehicle is driving after 300 metres of GPS silence.

The effect of calibration errors was examined on artificial data, giving a good understanding of the consequences of inaccuracies in each of the calibration angles. In future work, this will be exploited to make the system self-calibrating to an extent, using the offline map as a reference.

Another logical extensions would be implementation of a feedback loop of the trajectory to the vehicle suspension model to reduce perspective uncertainty. Also, the current version makes no use of the established consensus to refine the PUTs and recalculate the consensus. One can reasonably assume that such an iterative method will further improve the results, however at the cost of longer computation. Finally, a detailed analysis of the evolution of the uncertainties over time could be carried out to derive a confidence measure for the estimated position and orientation.

References

1. Amidi, O., Kanade, T., Miller, J.: Vision-based autonomus helicopter research at cmu. In: Proc. of Heli Japan 1998 (1998)

2. Azuma, T., Sugimoto, S., Okutomi, M.: Egomotion estimation using planar and non-planar constraints. In: Intelligent Vehicles Symposium (IV), 2010 IEEE. pp. 855–862 (2010)
3. Bouguet, J.: Visual Methods for Three-Dimensional Modeling. Ph.D. thesis, California Institute of Technology (May 1999)
4. Campbell, J., Sukthankar, R., Nourbakhsh, I., Pahwa, A.: A robust visual odometry and precipice detection system using consumer-grade monocular vision. In: Proc. of IEEE Int. Conf on Robotics and Automation (ICRA) 2005. pp. 3421–3427 (2005)
5. Cheng, Y., Maimone, M., Matthies, L.: Visual odometry on the mars exploration rovers. IEEE Robotics and Automation Magazine 13(2) (2006)
6. Comport, A., Malis, E., Rives, P.: Accurate quadrifocal tracking for robust 3d visual odometry. In: Proc. of IEEE Int. Conf on Robotics and Automation (ICRA) 2007. pp. 40–45 (2007)
7. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
8. Kitt, B., Geiger, A., Lategahn, H.: Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: Intelligent Vehicles Symposium (IV), 2010 IEEE. pp. 486–492 (2010)
9. Konolige, K., Agrawal, M., Sol, J.: Large-scale visual odometry for rough terrain. In: Int. Symposium on Research in Robotics (2007)
10. Levin, A., Szeliski, R.: Visual odometry and map correlation. In: Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition 2004. vol. 1-I, pp. 611–618 (2004)
11. Marks, R., Wang, H., Lee, M., Rock, S.: Automatic visual station keeping of an underwater robot. In: Proc. of IEEE Oceans 1994. pp. 137–142 (1994)
12. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. Image and Vision Computing 27(8) (2009)
13. Negahdaripour, S., Horn, B.: Direct passive navigation. IEEE Transactions on Pattern Analysis and Machine Intelligence 9(1) (1987)
14. Nistér, D.: An efficient solution to the five-point relative point problem. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(6) (2004)
15. Nistér, D., Naroditsky, O., Bergen, J.: Visual odometry for ground vehicle applications. Journal of Field Robotics 23 (2006)
16. Obdržálek, S., Matas, J.: A voting strategy for visual ego-motion from stereo. In: Intelligent Vehicles Symposium (IV), 2010 IEEE. pp. 382–387 (2010)
17. Scaramuzza, D., Fraundorfer, F., Siegwart, R.: Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In: Proc. of IEEE Int. Conf on Robotics and Automation (ICRA) 2009. pp. 4293–4299 (2009)
18. Tardif, J.P., Pavlidis, Y., Daniilidis, K.: Monocular visual odometry in urban environments using an omnidirectional camera. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2008. pp. 2531–2538 (2008)